# Smart Home Monitoring

**Student Name: Florian Poppinger Student ID: W20108867**

The Smart Home Monitoring System is designed to enhance home security and environmental monitoring through a network of sensors and devices integrated via a Raspberry Pi. The system captures real-time data on environmental conditions and detects motion, instantly notifying the homeowner with detailed information and images. The primary technologies used include the Raspberry Pi with a Sense HAT, a camera for motion detection, MQTT for messaging, Firebase for image storage, and Blynk for real-time interaction.

# 2. System Requirements

- **Hardware:**
  - Raspberry Pi 3B+ or newer
  - Sense HAT
  - Pi Camera
  - Stable internet connection
  - Power supply for Raspberry Pi
- **Software:**
  - Raspbian OS (or any compatible Raspberry Pi OS)
  - Python 3.6 or later
  - Libraries: paho-mqtt, PIL, numpy, firebase-admin, BlynkLib, smtplib, email, cryptography
  - Firebase account and setup
  - Blynk app with an account

# 3. Installation and Setup

**a. System Setup**

- Raspberry Pi Setup:
  - Install the Raspbian OS on your Raspberry Pi.
  - Connect the Sense HAT and Pi Camera to the Raspberry Pi.
  - Ensure the Raspberry Pi is connected to the internet.
- Dependency Installation:

    - Install required Python libraries using pip:
      *pip install paho-mqtt Pillow numpy firebase-admin BlynkLib cryptography*
- Firebase Setup:
    - Set up a Firebase project in the Firebase console.
    - Enable Firebase Storage.
    - Download the Firebase Admin SDK credentials file and place it in the project directory. (is not able to be pushed to git)

**b. Software Configuration**

- MQTT Broker Setup:
    - Register for an account with a broker like EMQX Cloud and get the broker details.
- Blynk Setup:
    - Create a new Blynk project to obtain an Auth token.
    - Set up virtual pins in Blynk app corresponding to the data types you are monitoring (temperature, humidity, pressure, etc.).
- Email Configuration:
    - Configure the SMTP settings in the Email_handler.py to match your email provider's requirements.

# 4. Code Overview

**a. Main Components**

- *Controller.py:*
    - Manages the main operational loop, threading, and integration of all components.
- *sensehat.py:*
    - Interfaces with the Sense HAT to read environmental data.
- *motion.py:*
    - Handles motion detection logic using the Pi Camera.
- *storeFileFB.py:*
    - Manages uploading images to Firebase and retrieving URLs.
- *blynk_handler.py:*
    - Provides real-time interaction and status updates through Blynk.
- *Email_handler.py:*
    - Sends email notifications with attachments and environmental data.
- *clients_sub.py, client_pub.py:*
    - Handle MQTT subscriptions and publications, respectively.

**b. Execution Flow**

- System initializes and starts separate threads for motion detection, data publication, and user control.
- On detecting motion via MQTT or upon user request via Blynk, images are captured, stored in Firebase, and emailed to the user.
- Blynk and MQTT facilitate real-time data updates and user interaction.

# 5. Usage Instructions

- Starting the System:
  - Run Controller.py to start the system:
    - *python Controller.py*
- Interacting with the System:
  - Use the Blynk app to view real-time data and control the system remotely.
  - Monitor your email for alerts and captured images from motion events.

# 6. Troubleshooting

MQTT Connection Failure: Ensure the broker details are correct and that your internet connection is stable.

Camera Not Working: Check the camera connection with the Raspberry Pi and verify File permissions in the OS.

Email Not Sending: Validate your SMTP settings and check the app password if using two-factor authentication.

# 8. Conclusion

This Smart Home Monitoring System offers a robust framework for home security and environmental monitoring, leveraging modern IoT technologies for efficient real-time updates and interactions. It is scalable, flexible, and can be enhanced with additional features as per user requirements.